Algorithmic advances for software radios

Matteo Frigo

Vanu Inc.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **21 MAY 2003**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Algorithmic Advances for Software Radios**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Vanu, Inc.**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
Also see ADM001473 , The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**11**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

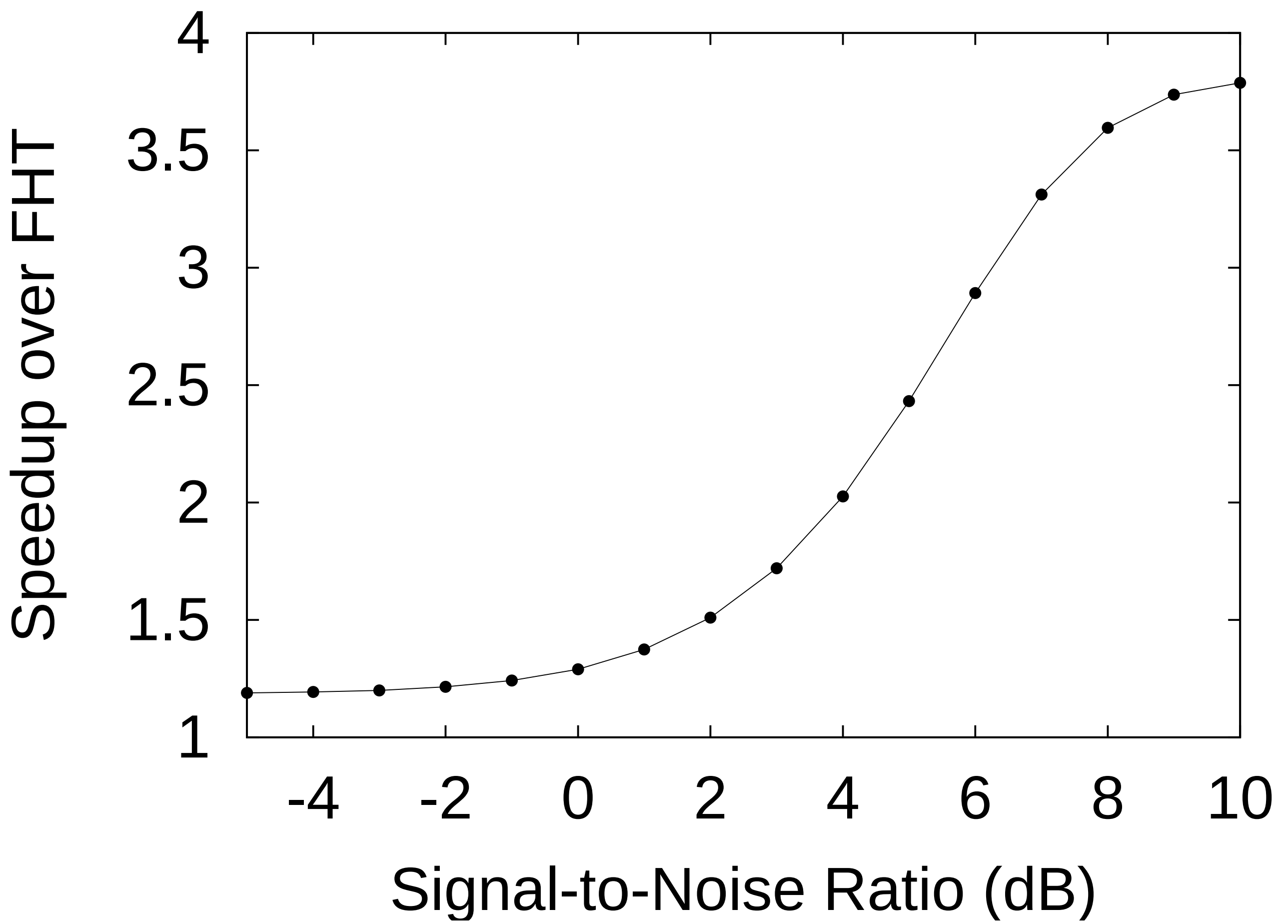
Software radios

- Replace most communication hardware with software.
- (E.g., our demo GSM basestation runs on COTS Pentium III PC.)
- Software radios require new algorithms.
- "Hybrid" CCK demodulator for 802.11b: up to 4 times faster than standard demodulator.
- "Lazy" Viterbi algorithm: up to 10 times than Viterbi for CDMA.

CCK demodulation

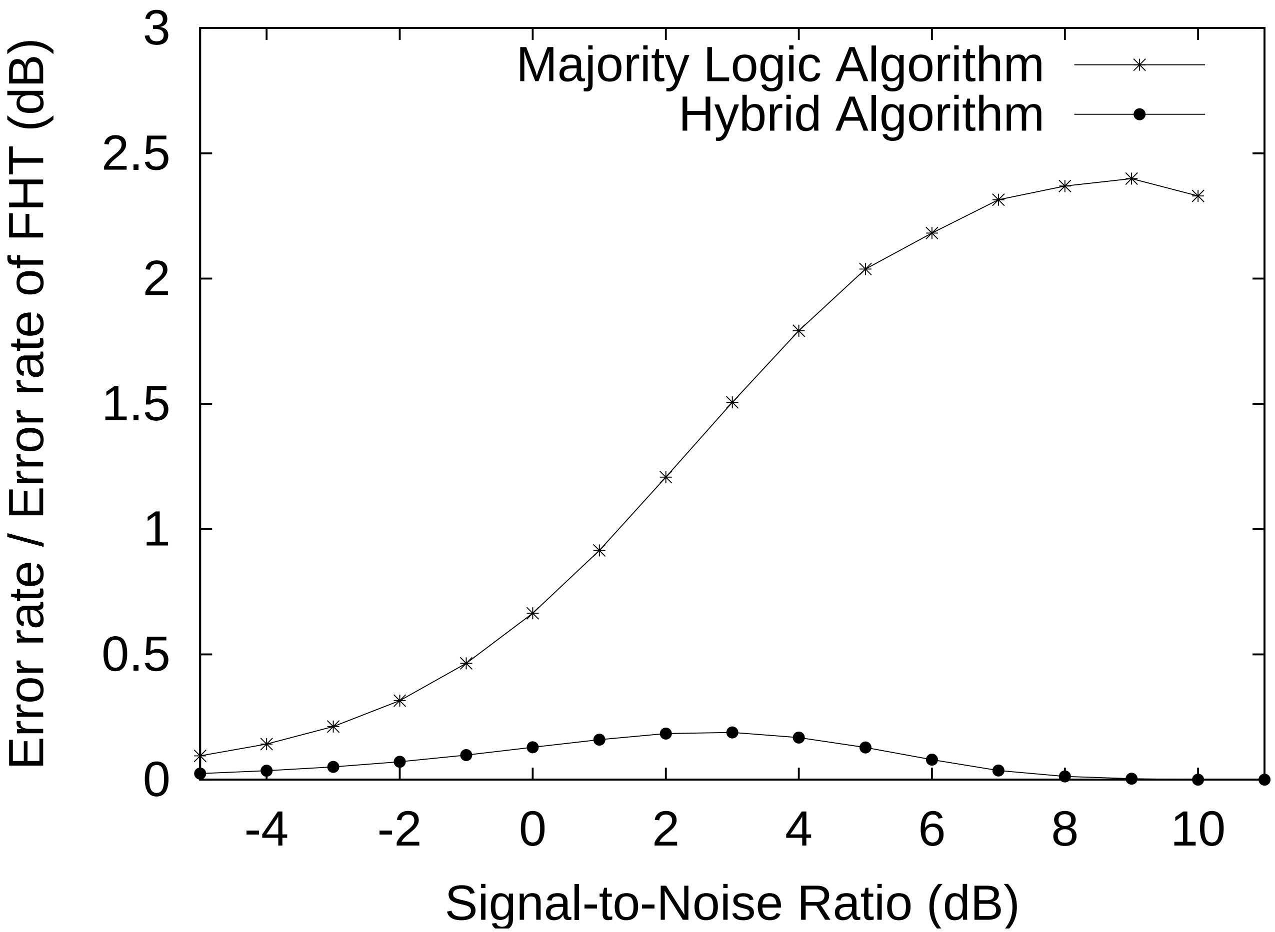
- Demodulation is the bottleneck in a software 802.11b implementation.
- Standard maximum-likelihood demodulator based on the fast Walsh-Hadamard transform (FHT).
- Majority-logic demodulators are efficient but suboptimal.
- Our *Hybrid* algorithm is almost as fast as majority logic and almost as “optimal” as the FHT.

Speed of Hybrid algorithm



Significantly faster than the fast Walsh-Hadamard transform.

Performance of Hybrid algorithm



Negligible loss of optimality ($\leq$ 0.2 dB).

CCK modulation

Input: 8 bits. **Output:** 8 "complex bits" (± 1 or $\pm i$, where $i = \sqrt{-1}$).

Map the 4 pairs of input bits into 4 complex bits ϕ_0, ϕ_1, ϕ_2, ϕ_3. Output vector $x(\phi)$, where:

$$
\begin{array}{rcllll}
x_0 & = & \phi_3 & & & \\
x_1 & = & \phi_3 & & & \phi_0 \\
x_2 & = & \phi_3 & & \phi_1 & \\
x_3 & = & \phi_3 & & \phi_1 & \phi_0 \\
x_4 & = & \phi_3 & \phi_2 & & \\
x_5 & = & \phi_3 & \phi_2 & & \phi_0 \\
x_6 & = & \phi_3 & \phi_2 & \phi_1 & \\
x_7 & = & \phi_3 & \phi_2 & \phi_1 & \phi_0
\end{array}
$$

CCK demodulators

- Maximum-likelihood: maximize correlation $|x(\phi) \cdot y|$ of received signal y with transmitted signal $x(\phi)$, over all ϕ. (Can be computed via fast Walsh-Hadamard transform.)
- Majority-logic: compute

$$
\begin{aligned}
\phi_0 &\approx x_1/x_0 \approx x_3/x_2 \approx x_5/x_4 \approx x_7/x_6 \; ; \\
\phi_1 &\approx x_2/x_0 \approx x_3/x_1 \approx x_6/x_4 \approx x_7/x_5 \; ; \\
\phi_2 &\approx x_4/x_0 \approx x_5/x_1 \approx x_6/x_2 \approx x_7/x_3 \; .
\end{aligned}
$$

The Hybrid algorithm

- Compute the quantities

$$\begin{aligned}
\bar{\phi}_0 &= (x_1 x_0^* + x_3 x_2^* + x_5 x_4^* + x_7 x_6^*) \; ; \\
\bar{\phi}_1 &= (x_2 x_0^* + x_3 x_1^* + x_6 x_4^* + x_7 x_5^*) \; ; \\
\bar{\phi}_2 &= (x_4 x_0^* + x_5 x_1^* + x_6 x_2^* + x_7 x_3^*) \; .
\end{aligned}$$

  Absent noise: the $\bar{\phi}_k$'s are purely real or imaginary.

- Let ϕ_k be the real or imaginary axis closest to $\bar{\phi}_k$.

- If $|\angle\phi_k - \angle\bar{\phi}_k| \leq \alpha$ we are done. (α is a magic number.)

- Otherwise, switch to maximum-likelihood demodulator.

Good value for $\alpha = \arctan(2/3) \approx 33.7°$.

Convolutional codes

Commonly used in TDMA/GSM cellular phones and other wireless standards.

Runtime of optimal decoders:

Algorithm	Best case	Worst case
Viterbi	$\Theta(2^k L)$	$\Theta(2^k L)$
A^*	$\Theta(L \log L)$	$\Theta(2^k L \log(2^k L))$
Our “Lazy Viterbi”	$\Theta(L)$	$\Theta(2^k L)$

(k = “constraint length”, L = input size.)

For A^* and Lazy Viterbi, runtime depends on SNR.

Convolutional decoders

Decoder	k	Pentium III cycles/bit	PowerPC 7400 cycles/bit	StrongARM cycles/bit
Lazy	6	201	200	226
Viterbi Optimized	6	316	239	310
Karn Unoptimized	6	1143	626	892
Lazy	7	205	203	232
Karn Optimized	7	558	486	641
Karn Unoptimized	7	2108	1094	1535
Karn SSE	7	108	N/A	N/A
Lazy	9	235	225	343
Karn Unoptimized	9	8026	3930	5561
Karn SSE	9	310	N/A	N/A

(k = constraint length.)

Conclusion

- Software radios require new algorithms.
- Noise-adaptive algorithms can save power and improve battery life.
- The flexibility of software radios allows the best algorithm to be chosen at run time.